UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/708,159	11/08/2000	Toshiaki Yasue	JP919990097US1	1032

7590 08/11/2006

William A Kinnaman Jr
IBM Corporation - MS P386
2455 South Road
Poughkeepsie, NY 12601

EXAMINER	
RUTTEN, JAMES D	

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 08/11/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Offic Action Summary	Application N .	Applicant(s)	
	09/708,159	YASUE ET AL.	
	Examin r	Art Unit	
	J. Derek Rutten	2192	

-- Th MAILING DATE of this communication appears n the c ver sheet with the correspondenc address --

Period f r Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 May 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-10* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-10* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Pri rity under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to Applicant's amendment dated 5/26/2006, responding to the 1/26/2006 Office action provided in the rejection of claims 1-10. No claims have been amended. Claims 1-10 remain pending in the application and have been fully considered by the examiner.

2. In the response filed 5/26/2006, Applicant essentially argues that the rejection under 35 U.S.C. 103(a) is improper since the Koblenz reference does not disclose loop optimization. This argument is not persuasive, and is further addressed below.

3. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Response to Arguments

4. On pages 6 and 7 of the response filed 5/26/2006, Applicant addresses the rejection of claims 1-10 under 35 U.S.C. § 112, first paragraph. Applicant limits the definition of a transfer

point in accordance with page 14 lines 29-30 in the originally filed specification as "a point at which program execution is transferred from an interpreted process to a loop process of a compiled code process." At the bottom of page 6 continuing to the presentation of figures on page 7, Applicant explains how a transfer point could be moved and/or copied. At the bottom of page 7, Applicant concedes that Aho depicts entry points to a loop, but asserts that this does not change the "thrust" of the previous argument. These arguments are persuasive. Accordingly, this rejection is withdrawn.

5. It is noted that Applicant has provided a definition of the term "transfer point" on page 6 of the response filed 5/26/2006 as "a point at which program execution is transferred from an interpreted process to a loop process of a compiled code process." It is further noted that in the on page 7 of the response filed 4/1/2004, Applicant asserted that transfer points are well known in the art as shown in Section 10.4 (pages 602-608) of Aho et al.

6. On page 8, Applicant addresses the rejection of claims 1-10 under 35 U.S.C. § 112, second paragraph. Applicant attempts to clarify the invention by asserting that the movement of a transfer point does not involve the movement of code. This argument relates to a comment at the bottom of page 6 suggesting "[o]ne might 'move' a transfer point by changing the target address of a transfer instruction so that it points to the top of a loop rather than inside the loop." This argument is persuasive. Accordingly, this rejection is withdrawn.

7. Applicant addresses the rejection of claims 1-10 under 35 U.S.C. § 103(a) on pages 8-10. Applicant particularly addresses USPN 5530866 to Koblenz et al. ("Koblenz"), and essentially argues that Koblenz is directed to manipulation of an abstract flow graph for the purpose of assigning physical registers, and not the modification of a loop process for the purpose of code

optimization. It is agreed that Koblenz teaches the manipulation of an abstract flow graph. As shown by Applicant on page 7 of the response filed 5/26/2006, manipulation of an abstract flow graph is helpful in loop optimization. Regardless of Koblenz' use of loop manipulation for physical register assignment, the teaching of loop manipulation in an abstract flow graph provides valuable information regarding the nature of domination and irreducibility. Further, Koblenz is not relied upon for loop optimization, rather such optimizations are addressed by Aho (e.g. at least Aho Section 10.4; also pages 679-680). As such, Applicant's argument is not persuasive.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 1-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over prior art of record "Compilers: Principles, Techniques, and Tools" by Aho et al. (hereinafter "Aho") in view of prior art of record U.S. Patent 6,513,156 to Bak et al. (hereinafter referred to as "Bak"), further in view of prior art of record "Compiler Transformations for High-Performance Computing" by Bacon et al. (hereinafter referred to as "Bacon"), further in view of U.S. Patent 5,530,866 to Koblenz et al. (hereinafter "Koblenz").

In regard to claim 1, Aho discloses:

A program execution method ... comprising the steps of: optimizing the loop process, said optimizing step including the steps of: copying code from the top of the loop process to a point that post-dominates said top of said loop process and said one or more ...points to a location immediately preceding said loop process if said ... points are located inside said loop process; Aho discusses code optimizations that benefit from the reducibility of flow graphs containing loops in terms of using intervals, interval graphs, and node splitting (see pages 664-668). In particular, node splitting is a technique that is used to produce a limit flow graph of a single node (page 666, last paragraph). Additional nodes are created that precede the original node. Fig. 10.49 shows a flow graph with 3 nodes with edges from 1 to 2, 1 to 3, 2 to 3, and 3 to 2. The loop of nodes 2 and 3 show two incoming edges. Node 2 is split into nodes 2a and 2b. When combined with node 1, node 2a now precedes the loop that is formed between nodes 2b and 3. Further description is found on pages 679-680, which describes duplicating a region that represents a node, and placing that region in a location preceding the node. This process is shown in Fig. 10.57 on page 680.

Aho further discloses common subexpression elimination (Fig. 10.7 on page 593), and code motion (page 596). Also, as asserted by Applicant (see 4/1/04, page 7), Aho discloses the notion of transfer points (Section 10.4, pages 602-608).

Aho does not expressly disclose: moving transfer points to the top of a loop process; transferring a method from an interpreter process to a compiled code process;

storing information for generating recalculation code for specific transfer points; performing a recalculation during a transfer process; or privatization.

However, in an analogous environment, Koblenz teaches: *moving said one or more ...points to the top of a loop process if they can be moved there without a problem occurring*; See Koblenz column 8 lines 23-28:

> Irreducible loops do not exhibit a loop top; however, all basic blocks in an irreducible loop that are reached by a forward control flow edge from a basic block outside the loop can be combined into a single summary loop top in constructing the tile tree. This summary node will dominate every basic block in the loop.

Also, in an analogous environment, Bak teaches: *transferring, from an interpreter process to a compiled code process, a method that is currently being executed for code that includes a plurality of transfer points at which program execution is transferred from the interpreter process to the compiled code process <via one of said transfer points>*. See column 2 lines 40-45:

> the hybrid virtual and native machine instructions may be easily transformed back to the original virtual machine instructions, and the flexibility of compiling only certain portions of a function into native machine instructions allows for better optimization of the execution of the function.

storing information for generating recalculation code for one or more specific transfer points See column 2 line 65 – column 3 line 1:

> A copy of a selected virtual machine instruction at a beginning of the portion of the function is **stored** and a back pointer to a location of the selected virtual machine instruction is also stored.

and performing a recalculation during a transfer process See column 3 lines 1-5:

> The selected virtual machine instruction is overwritten with a new virtual machine **instruction that specifies execution** of the native machine instructions so that the function includes both virtual and native machine instructions.

Bak is generally concerned with mixed execution of interpreted and compiled code sequences where the compiled code process is referred to as a "snippet", and transfer

points to the compiled code process are indicated with a "go--native" instruction (column 6 lines 31-35 and column 7 lines 28-37).

Also in an analogous environment, Bacon teaches: *privatization* See page 395 Section 7.1.3:

> When a scalar is used within a loop solely as a scratch variable, each processor can be given a private copy so the use of the scalar need not involve any communication.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Koblenz' moving of transfer points with Bacon's optimizations with Bak's mixed mode interpreter in Aho's code optimizer. One of ordinary skill would have been motivated to remove transfer points from a loop in order to make them reducible since many optimizations depend on reducibility (Aho page 607). Further, one would have been motivated to transfer the execution of an interpreted loop to natively compiled instructions since native code executes more quickly than interpreted code.

As per claim 2, the above rejection of claim 1 is incorporated. Aho does not expressly disclose choosing transfer points for transferring from interpreted mode to compiled mode execution.

However, Bak teaches *defining as a new transfer point, a point from said interpreter process to said compiled code process whereat, when said method that is currently being executed is replaced, the execution speed is increased compared with when said method is not replaced* (column 6 line 61 – column 7 line 5).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Bak's selection of transfer points in O'Brien's code optimizer. One of ordinary skill would have been motivated to improve code so that a program will execute in less time.

As per claim 3, the above rejections of claims 1 and 2 are incorporated. Aho does not expressly disclose generating, storing, or employing information for transferring execution from interpreted to compiled execution.

However, Bak teaches:

generating information required to perform a transfer from said interpreter process to said compiled code process (column 7 lines 28-40)*; and*

storing said generated information while correlating said generated information with said transfer points (column 7 lines 28-40 as cited above),

wherein, at said recalculation step, said information stored for said transfer points is employed (column 7 lines 63-67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Bak's transfer information with O'Brien's code optimizer. One of ordinary skill would have been motivated to enable the transfer of interpreted execution to natively compiled execution, which is necessarily supported by information regarding the location of code, to increase the speed of a program.

As per claim 4, Aho does not expressly disclose a program storage device.

However, Bak teaches the use of a program storage device to hold program instructions (column 4 lines 46-50).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Bak's program storage device with O'Brien's code optimizer. One of ordinary skill would have been motivated to store copies of a program on media that enables the distribution of the program to colleagues or customers.

All further limitations have been addressed in the above rejection of claim 1.

10. Claims 5, 6, 8, and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bak, in view of Koblenz and Aho.

In regard to claim 5, all limitations have been addressed in the above rejection of claim 1. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Koblenz' movement of transfer points, and Aho's optimization with Bak's transfer of execution. One of ordinary skill would have been motivated to make a loop reducible in order to better optimize it (Bak column 2 lines 24-29, Aho first paragraph of section 10.9 on page 660, and Koblenz column 8 lines 15-23).

In regard to claim 6, all limitations have been addressed in the above rejection of claim 1. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Aho's code copying with Bak's transfer process in order to facilitate the reducibility of a graph which would allow better optimization.

In regard to claim 8, all limitations have been addressed in the above rejections of claims 4 and 5.

In regard to claim 9, the above rejection of claim 8 is incorporated. All further limitations have been addressed in the above rejection of claim 1.

11. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Aho in view of Bak.

In regard to claim 7, all limitations have been addressed in the above rejection of claim 1.

In regard to claim 10, all limitations have been addressed in the above rejection of claims 1 and 4.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to J. Derek Rutten whose telephone number is (571)272-3703. The examiner can normally be reached on T-F 6:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571)272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

jdr

TUAN DAM
SUPERVISORY PATENT EXAMINER